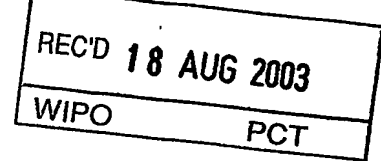


12 2 Feb 2003

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)



**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen:

102 40 770.3

Anmeldetag:

30. August 2002

Anmelder/Inhaber:

Systemonic AG, Dresden/DE

Bezeichnung:

Verfahren zur Initialisierung von programmierbaren
Systemen

IPC:

G 06 F 9/445

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ur-
sprünglichen Unterlagen dieser Patentanmeldung.

München, den 8. Juli 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Faust



LIPPERT, STACHOW, SCHMIDT & PARTNER
Patentanwälte · European Patent Attorneys · European Trademark Attorneys
Krenkelstraße 3 · D-01309 Dresden
Telefon +49 (0) 3 51.3 18 18-0
Telefax +49 (0) 3 51.3 18 18 33

Ad-ak/ak

30. August 2002

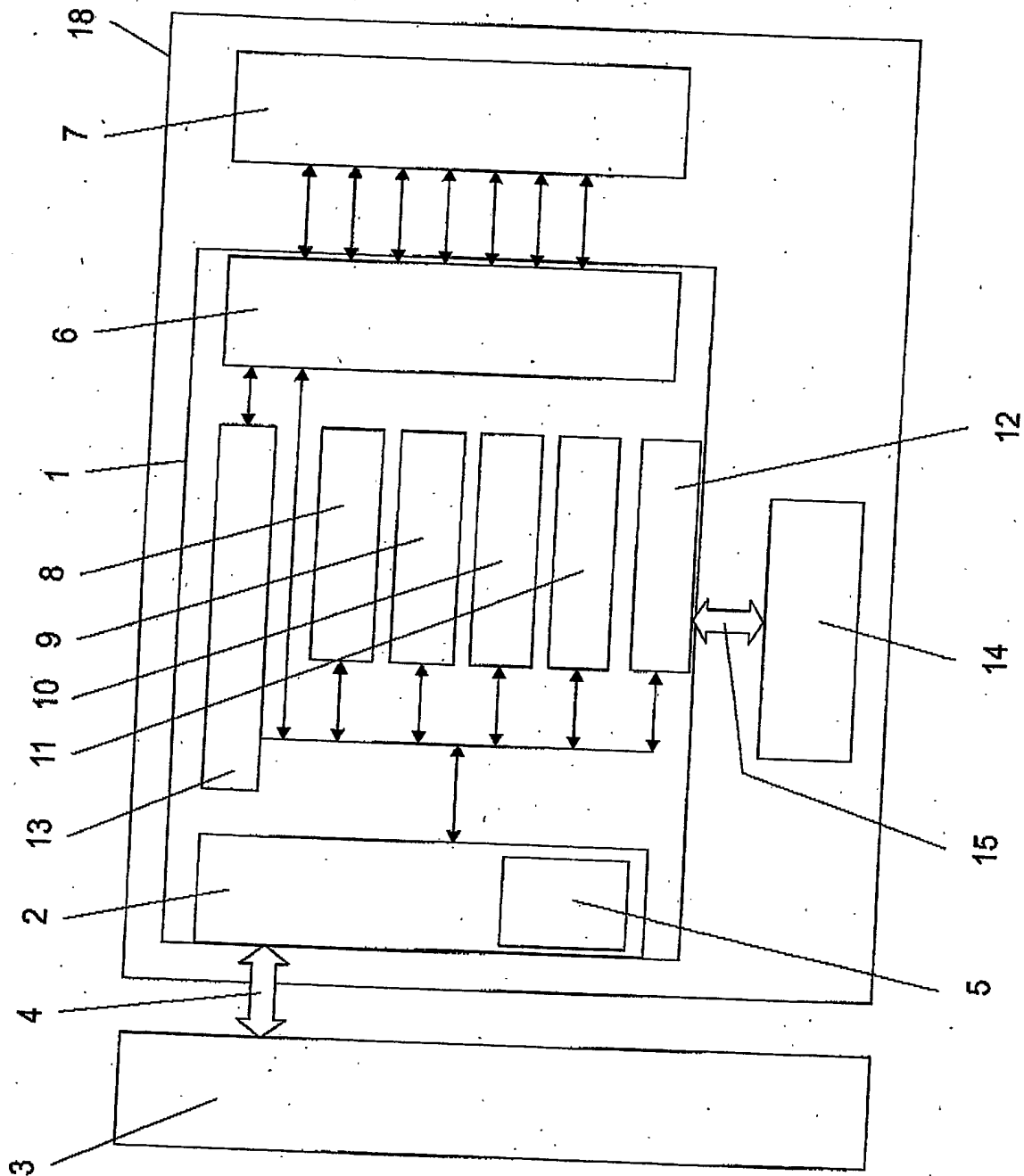
5 **Systemonic AG**
01099 Dresden

10 **Verfahren zur Initialisierung von programmierbaren Systemen**

Zusammenfassung

15 Die Erfindung, die ein Verfahren zur Initialisierung von programmierbaren Systemen betrifft, bei dem die zur Initialisierung von Registern und internen und/oder externen Baugruppen notwendigen Informationen in einem externen Speicher enthalten sind und ausgelesen werden, liegt die Aufgabe zugrunde, ein Verfahren anzugeben, bei dem alle Register und
20 Baugruppen initialisierbar sind, eine flexible Initialisierung der externen Elektronik möglich ist, die ASIC-Entwicklung vereinfacht wird und unterschiedliche EEPROM-Typen unterstützt werden. Gemäß der Erfindung wird die Aufgabe dadurch gelöst, dass nach dem Einschalten oder einem anderen, einen Neustart
25 auslösenden Ereignis, gesteuert durch ein Programm eines Instruktionsspeichers, ein Transfer von Initialisierungsinformationen von einem externen oder internen nichtflüchtigen Speichermedium zu einem internen Speicher erfolgt, dass die Initialisierungsinformationen
30 Initialisierungsdaten und/oder mindestens ein Initialisierungsprogramm enthalten, dass die Initialisierung der Register und Baugruppen durch ein oder mehrere im System angeordnete Prozessorelemente oder andere intelligente Baublöcke gesteuert erfolgt, die ihrerseits durch das
35 Initialisierungsprogramm gesteuert werden. (Fig. 2)

Fig. 2



LIPPERT, STACHOW, SCHMIDT & PARTNER
Patentanwälte · European Patent Attorneys · European Trademark Attorneys
Krenkelstraße 3 · D-01309 Dresden
Telefon +49 (0) 3 51 3 18 18-0
Telefax +49 (0) 3 51 3 18 18 33

Ad-ak/ak

30. August 2002

5 **Systemonic AG**
01099 Dresden

10 **Verfahren zur Initialisierung von programmierbaren Systemen**

Die Erfindung betrifft ein Verfahren zur Initialisierung von programmierbaren Systemen bei dem die zur Initialisierung von Registern und internen und/oder externen Baugruppen notwendigen
15 Informationen in einem externen Speicher enthalten sind und ausgelesen werden, insbesondere zur Anwendung in programmierbaren System-on-Chip ASIC Elementen.

Komplexe mikrocontrollergestützte elektronische Systeme, insbesondere im Bereich der Personal-Computer-Peripherie setzen sich zumeist aus produktionstechnischen und wirtschaftlichen Gründen aus wenigen Bauteilen zusammen. Im allgemeinen wird versucht, sämtliche Logik auf einen applikationsspezifischen, integrierten Baustein (ASIC = Application specific integrated
20 circuit) unterzubringen. Dieser ASIC enthält die Interfaces, die die Verbindung mit dem PC herstellen, zum Beispiel PCI- (PCI = Peripheral Components Interconnect), Cardbus- oder USB- Controller (USB = Universal Serial Bus). Aufgrund des Charakters der Produkte, welche in hohen Stückzahlen zu kleinen
25 Preisen gefertigt werden, ist es üblich, bei Halbleiter-Designhäusern eine hartverdrahtete Lösung zu entwickeln, die bei Vertragspartnern hergestellt und an OEM-Kunden (OEM = Originalgerätehersteller) verkauft wird, welche die Halbleiter in ihre Peripheriegeräte integrieren. Die sich
30 ergebenden Endprodukte unterscheiden sich dabei zumeist durch

Funktionsumfang, Leistung, Stromverbrauch und Preis, je nachdem welche Beschaltung und zusätzlichen Bauteile für die jeweilige Lösung eingesetzt wurde.

- 5 Durch diese schaltungstechnischen Unterschiede müssen die verschiedenen Hardwaregeräte softwaremäßig unterschiedlich angesprochen und behandelt werden. Daran angepasste Treiber auf Betriebssystemseite sind notwendig und oftmals auch von den OEM-Kunden erwünscht, um sich dadurch weiter von
- 10 Konkurrenzprodukten abzugrenzen.

Darüber hinaus muss die Zusatzelektronik auf der Platine schon beim Einschalten oder Einstecken auf eine variable Art und Weise in einen Initialzustand gebracht werden.

- 15 Aus den beschriebenen Forderungen ergeben sich zwei Problemstellungen:

- 20 Zum einen muss eine Erkennung für jedes externe Gerät durchgeführt und ein entsprechender Gerätetreiber zugeordnet werden. Moderne PC-Betriebssysteme müssen während dem Betrieb (z. B. USB, Cardbus) oder vor dem Betrieb (z. B. PCI) hinzugefügte externe Baugruppen automatisch identifizieren und einen passenden Treiber zuordnen können. Wenn die Merkmale,
- 25 anhand derer eine Identifizierung für das Betriebssystem möglich ist, jedoch ausschließlich auf dem ASIC untergebracht werden, so ist eine Unterscheidung zwischen verschiedenen OEM-Produkten unmöglich.
- 30 Zum anderen muss eine Initialisierung der OEM-spezifischen Elemente beim Einschalten durchgeführt werden. Es kann vorkommen, dass die externen Elektronikbaugruppen, die sich neben dem ASIC auf dem Gerät befinden, direkt beim Einschalten bzw. Einstecken oder innerhalb kürzester Zeit danach in einen
- 35 definierten Ausgangszustand gebracht werden müssen, um z. B. eine Zerstörung, übermäßigen Stromverbrauch oder verwirrende

Statusanzeigen, durch Leuchtdioden oder ein Display, zu vermeiden. Da der ASIC in der Schaltung meistens der einzige „intelligente“ Baustein ist, der diese Aufgabe ausführen kann, die externe Beschaltung sich aber im allgemeinen, wie oben beschrieben, unterscheidet, muss ein Weg gefunden werden, dem ASIC die Informationen über die Initialisierung der anderen Baugruppen zu vermitteln, damit dieser die Initialisierung dementsprechend vornehmen kann.

- 10 Bekannte Verfahren basieren darauf, einen kleinen EEPROM (= Electrically Erasable Programmable Read Only Memory - elektrisch lösch- und programmierbarer Nur-Lese-Speicher) auf der Geräteplatine unterzubringen und mit dem ASIC über einen meist seriellen Bus zu verbinden. Im EEPROM sind
- 15 Identifikationsmerkmale und Seriennummer gespeichert. Unter den Identifikationsmerkmalen versteht man beispielsweise bei PCI/Cardbus die Produkt-ID, Vendor-ID, Subsystem-ID und Subsystem Vendor-ID als Revisionskennung und Geräteklasse. Als Beispiel für eine Seriennummer soll die MAC-ID bei Ethernet-
- 20 Netzwerkkarten stehen.

Diese OEM-spezifischen Informationen werden beim Einschalten des Gerätes durch den Endverbraucher mittels einer Hardwarelogik aus dem EEPROM ausgelesen und in die

25 entsprechenden Register im PCI/USB-Kern des ASIC transferiert. Dadurch kann das Gerät vom Rechner erkannt werden.

Die Register in den IO-Blocks, über die die restliche Elektronik auf dem Gerät gesteuert wird, bleiben entweder

30 uninitialisiert, werden auf einen fest im ASIC „eingebrannten“ Reset-Wert eingestellt oder können durch die Hardwarelogik nach dem EEPROM Inhalt gesetzt werden.

Diese bekannte Lösung weist die im folgenden näher

35 beschriebenen Nachteile auf.

Zum einen ist die dazu notwendige Hardware relativ komplex. Die Logik muss im ASIC hartverdrahtet implementiert werden. Speziell mit komplizierten Busprotokollen, wie I²C, ist die Verifizierung und die Implementierung der Logik (zusätzliche Gatter, Transistoren, Fläche auf dem Silizium-Die) aufwendig.

Zum anderen ist das existierende Verfahren nicht flexibel. Bereits während der Chipdesignphase muss festgelegt werden, welche Register im ASIC (Adresse) wann und mit welchem Inhalt (Datum) aus dem EEPROM beschrieben (initialisiert) werden müssen. Die oben beschriebene Logik muss entsprechend diesen Erfordernissen aufgebaut werden.

Der Erfindung liegt somit die Aufgabe zugrunde, ein Verfahren anzugeben, bei dem alle Register und Baugruppen initialisierbar sind, eine flexible Initialisierung der externen Elektronik möglich ist, die ASIC Entwicklung vereinfacht wird und unterschiedliche externe und interne Speichermedien unterstützt werden.

Gemäß der Erfindung wird die Aufgabe mit einem Verfahren zur Initialisierung von programmierbaren Systemen der eingangs genannten Art dadurch gelöst, dass nach dem Einschalten oder einem anderen, einen Neustart auslösenden Ereignis, gesteuert durch ein Programm eines Instruktionsspeichers, ein Transfer von Initialisierungsinformationen von einem externen oder internen nichtflüchtigen Speichermedium zu einem internen Speicher erfolgt, dass die Initialisierungsinformationen Initialisierungsdaten und/oder mindestens ein Initialisierungsprogramm enthalten, dass die Initialisierung der Register und Baugruppen durch ein oder mehrere im System angeordnete Prozessorelemente oder andere intelligente Baublöcke gesteuert erfolgt, die ihrerseits durch das Initialisierungsprogramm gesteuert werden.

Erfindungsgemäß wird die Initialisierung der Register und

Baugruppen durch einen oder mehrere Prozessorelemente durchgeführt. Dieser benötigt nach dem Einschalten oder nach einem externen Re-Start Ereignis des Gerätes ein abzuarbeitendes Programm. Das Programm für den Start der Initialisierungsphase ist in einem sogenannten Instruktionsspeicher (Bootstrap-Loader) enthalten. Dieses Programm steuert die Übertragung der Initialisierungsinformationen von einem externen EEPROM in einen RAM-Speicher (Instruction und/oder Data RAM) (RAM = Random Access Memory). Dabei können die Initialisierungsinformationen sowohl Initialisierungsdaten als auch ein Initialisierungsprogramm beinhalten. Initialisierungsdaten sind Identifikationen (ID), wie beispielsweise die Product-ID, Vendor-ID, Subsystem Vendor-ID oder eine Seriennummer einer Ethernet-Netzwerkkarte. Das Initialisierungsprogramm steuert das Prozessorelement nach der Übertragung der Initialisierungsinformationen in den Instruction-RAM und realisiert die Initialisierung der Register und Baugruppen.

In einer Ausgestaltung der Erfindung ist vorgesehen, dass nach dem Transfer eine Integritätsprüfung der Initialisierungsinformationen erfolgt und dass, gesteuert durch das Ergebnis der Integritätsprüfung, eine Programmverzweigung durchgeführt wird.

In einer Ausführung der Erfindung ist vorgesehen, dass beim Erkennen eines falschen oder fehlenden internen oder externen Speichermediums, eine Fehlerroutine abgearbeitet wird, die die Initialisierung mit Standardwerten durchführt oder den Inhalt des internen oder externen Speichermediums ganz oder teilweise wiederherstellt.

Gleichfalls ist denkbar, dass die Informationen als ausführbares Macro-Programm vorliegen und vom Prozessorelement interpretiert werden. Mischformen aus beiden Verfahren sind

ableitbar.

- Nach der Übertragung der Initialisierungsinformationen in den Instruction-RAM erfolgt eine Integritätsprüfung der Daten, beispielsweise durch die Ermittlung einer Prüfsumme. In Abhängigkeit des Ergebnisses wird durch das Prozessorelement entweder das gerade übertragene Initialisierungsprogramm oder Macro-Anweisungen abgearbeitet oder eine Routine zur Ausnahmefallbehandlung im Instruktionsspeicher angesprungen.
- 10 Durch diese Routine werden die funktional wichtigsten Register im ASIC so programmiert, dass ein Ansprechen des Gerätes über die jeweilige PC-Schnittstelle zumindest grundsätzlich möglich ist. Ist in der Einschaltphase ein fehlendes oder falsches Speichermedium erkannt worden, wird ebenfalls eine Routine zur
- 15 Initialisierung mit Standardwerten abgearbeitet.

- In einer weiteren Ausgestaltung der Erfindung ist vorgesehen, dass die Initialisierungsdaten als Standardwerte aus dem Speichermedium gelesen werden, durch das Prozessorelement
- 20 verändert und die veränderten Initialisierungsdaten zur Initialisierung verwendet werden.

- Im EEPROM sind Standardwerte zum Beispiel für die Product-ID, Vendor-ID, Subsystem Vendor-ID oder eine Seriennummer einer
- 25 Ethernet-Netzwerkkarte u. a. abgespeichert. Diese Werte können direkt zur Initialisierung der Register und/oder Baugruppen verwendet werden. Durch das Prozessorelement kann aber auch eine Veränderung der Standardwerte oder eine durch ein Ereignis gesteuerte Alternativauswahl vorgenommen werden. Unter dem
- 30 Aspekt der Fehlerbehandlung ergibt sich hieraus die Möglichkeit, mit Hilfe der Supportlogik die Initialisierungsinformationen des externen EEPROM zu verändern oder wiederherzustellen, wobei auch eine Neuberechnung der Prüfsumme erfolgen kann.

35

- In einer Ausgestaltungsform der Erfindung ist vorgesehen, dass

durch das Initialisierungsprogramm von dem Prozessorelement Initialisierungsdaten berechnet und zur Initialisierung verwendet werden.

5 Das Prozessorelement kann, durch das Initialisierungsprogramm gesteuert, Initialisierungsdaten, beispielsweise in Abhängigkeit des Zustandes eines Ports oder Registers, berechnen.

10 In einer Ausführung der Erfindung ist vorgesehen, dass in Abhängigkeit von Zustandsdaten von Peripheriekomponenten und/oder internen Komponenten deren Initialisierungsdaten und die Daten für die internen Komponenten berechnet werden.

15 Der Programmablauf des Initialisierungsprogramms kann so gestaltet werden, dass in Abhängigkeit von Zuständen einzelner interner oder externer Register oder Baugruppen ein Initialisierungswert gewählt oder errechnet wird. Dazu wird beispielsweise in einem ersten Schritt eine Abfrage eines Registers oder eines Ports realisiert und in Abhängigkeit dieses Ergebnisses erfolgt die Initialisierung nach einem Sprung an eine dafür vorgesehene Stelle im Programmablauf.

20 In einer besonderen Ausführung der Erfindung ist vorgesehen, dass das Prozessorelement nach erfolgter Initialisierung in einen Stromsparmodus übergeht.

30 Nach erfolgter Initialisierung besteht die Möglichkeit das Prozessorelement in einen Stromsparmodus zu versetzen, aus dem er beispielsweise durch ein Signal von einem PC oder einer Peripheriebaugruppe zurückversetzt wird.

35 In einer besonderen Ausführungsform der Erfindung ist vorgesehen, dass die Initialisierung weiterer Prozessorelemente gestartet und überwacht wird.

Das Prozessorelement kann die Initialisierung weiterer im System vorhandener Prozessoren vornehmen, die nachfolgend ihre eigenen Initialisierungsroutinen starten. Die Beendigung der Initialisierung wird entweder an den ersten Prozessor zurückgemeldet oder der erste Prozessor übergibt die Steuerung an einen anderen Prozessor.

In einer weiteren Ausführungsform der Erfindung ist vorgesehen, dass eine Anpassung an verschiedene Speichermedien erfolgt.

Der im ASIC angeordnete Instruktionsspeicher enthält ein nach dem Einschalten des Gerätes abzuarbeitendes Start-Programm und realisiert das Auslesen der Initialisierungsinformationen aus dem externen EEPROM und die Übertragung in den Instruction-RAM. Dieses Start-Programm enthält eine Routine, die das angeschlossene Speichermedium erkennt und für die Einhaltung des jeweils notwendigen Übertragungsprotokolls sorgt.

In einer weiteren Ausgestaltungsform der Erfindung ist vorgesehen, dass das Initialisierungsprogramm weitere Daten und/oder Programmcode aus einem Speichermedium nachlädt.

Bei Erreichen eines bestimmten Zustandes oder Ereignisses kann das Prozessorelement weitere Initialisierungsinformationen (Programmcode oder Zustandsdaten) nachladen.

Die Erfindung soll nachfolgend anhand eines Ausführungsbeispiels näher erläutert werden. In den zugehörigen Zeichnungen zeigt

Fig. 1 eine Schaltungsanordnung nach dem Stand der Technik und

Fig. 2 eine Schaltungsanordnung zur Umsetzung des erfindungsgemäßen Verfahrens.

Zur Realisierung des erfindungsgemäßen Verfahrens wurden in einem ASIC 1 neben dem aus dem Stand der Technik bekannten Bus-Interface 2, über das die Koppelung an einen PC 3 durch ein erstes Bussystem 4 erfolgt und welches zu initialisierende ID-Register 5 beinhaltet, und dem I/O-Interface 6, das die Koppelung zur externen Elektronik 7 realisiert, folgende Bestandteile integriert. Ein Prozessorelement 8, welches im Stand der Technik nicht unbedingt ein Bestandteil der ASIC-Logik 1 sein muss, ein Instruktionsspeicher 9 (Bootstrap-Loader), ein Instruction-RAM 10, ein Daten-RAM 11, ein EEPROM-Interface 12 und eine Support-Logik 13. Der aus dem Stand der Technik bekannte externe EEPROM 14 ist ebenfalls außerhalb des ASIC 1 angeordnet und über ein zweites Bussystem 15 mit diesem verbunden. Als zweites Bussystem 15 kann ein I²C, SPI oder Microwire-Bus zum Einsatz kommen. Der ASIC 1, das externe EEPROM 14 und die externe Elektronik 7 bilden ein durch den PC 3 zu steuerndes Peripheriegerät 18.

Der im ASIC angeordnete Instruktionsspeicher 9 enthält ein nach dem Einschalten des Gerätes abzuarbeitendes Start-Programm und realisiert das Auslesen der Initialisierungsinformationen aus dem externen EEPROM 14 und die Übertragung in den Instruction-RAM 10. Da der Austausch der Bussignale vom jeweilig genutzten externen EEPROM 14 und Bussystem 15 abhängig ist und der Vorgang durch ein Programm gesteuert wird, ist die Anpassung an verschiedene Bussysteme 15 oder EEPROM-Typen 14 programmtechnisch realisierbar. Damit besteht auch die Möglichkeit eine automatische Erkennung des angeschlossenen EEPROM-Typs 14 zu realisieren.

Für die Entwicklung des Instruktionsspeicher 9 können Standard-Entwicklungs- und Debuggingwerkzeuge anstelle teurer Spezialwerkzeuge eingesetzt werden.

Die zu übertragenden Initialisierungsinformationen bestehen aus den Initialisierungsdaten und einem Initialisierungsprogramm.

Nach der Übertragung der Initialisierungsinformationen in den Instruction-RAM 10 wird eine Integritätsprüfung zur Sicherstellung der fehlerfreien Übertragung der Daten durchgeführt.

Wird bei dieser Überprüfung beispielsweise anhand der Prüfsumme festgestellt, dass die Übertragung fehlerfrei war, erfolgt die weitere Programmausführung unter Nutzung des gerade in den Instruction-RAM 10 übertragenen Initialisierungsprogramms. Dieses Programm führt die eigentliche Initialisierung der Register 5 und Baugruppen unter Zuhilfenahme der Initialisierungsdaten durch. Durch diese Initialisierung werden beispielsweise die notwendigen Einstellungen im Bus-Interface 2 vorgenommen, um die Kommunikation mit dem PC 3 sowie die Initialisierung der anderen internen Baugruppen und des I/O-Interfaces 6 für die Steuerung der externen Elektronik 7 zu ermöglichen.

Ergibt die Integritätsprüfung einen Fehler, weil beispielsweise die Prüfsumme inkorrekt oder gar kein EEPROM 14 angeschlossen ist, so kann eine Routine zur Ausnahmefallbehandlung im Instruktionsspeicher 9 angesprungen werden, die die funktional wichtigsten Register im ASIC 1 so programmiert, dass ein Ansprechen des Gerätes über die jeweilige PC-Schnittstelle 4 zumindest grundsätzlich möglich ist. Durch diese Grundeinstellung und die im ASIC 1 integrierte Support -Logik 13 besteht die Möglichkeit ein defektes Gerät durch Neuprogrammierung des EEPROM 14 mit Neuberechnung der zugehörigen Prüfsumme zu reparieren oder die initiale Programmierung in der Produktionsstätte durchzuführen.

Durch die programmgesteuerte Initialisierung ist es möglich, alle vom Prozessorelement 8 direkt oder indirekt adressierbaren Register 5 oder Baugruppen und davon abhängige Zustandsautomaten zu setzen. Des weiteren besteht die

Möglichkeit die Initialisierung dynamisch, dass heißt in Abhängigkeit von Eingangswerten, durchzuführen. Damit ist es möglich, sofort in der Inbetriebnahmephase des Gerätes, in einem Display 7 anzuzeigen, ob eine bestimmte Bedingung erfüllt ist oder nicht.

Darüber hinaus ermöglicht das erfindungsgemäße Verfahren dem Peripheriegerät 18 ein eigenständiges Agieren um den PC 3 aus einem initialen Ruhezustand „aufzuwecken“ (z.B. Wake-on-Lan oder Wake-up bei Faxanruf).

LIPPERT, STACHOW, SCHMIDT & PARTNER

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51 3 18 18-0

Telefax +49 (0) 3 51 3 18 18 33

Ad-ak/ak

30. August 2002

5 **Systemonic AG**
01099 Dresden

10 **Verfahren zur Initialisierung von programmierbaren Systemen**

Bezugszeichenliste

- | | |
|----|----------------------------------|
| | 1 ASIC |
| 15 | 2 Businterface |
| | 3 Personalcomputer (PC) |
| | 4 erstes Bussystem |
| | 5 ID-Register |
| | 6 I/O-Interface |
| 20 | 7 externe Elektronik |
| | 8 Prozessorelemente |
| | 9 Instruktionsspeicher |
| | 10 Instruction-RAM |
| | 11 Daten-RAM |
| 25 | 12 EEPROM-Interface |
| | 13 Support-Logik |
| | 14 Speichermedium (z. B. EEPROM) |
| | 15 zweites Bussystem |
| | 16 Initialisierungssteuerung |
| 30 | 17 Hauptfunktionslogik |
| | 18 Peripheriegerät |

LIPPERT, STACHOW, SCHMIDT & PARTNER

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51 3 18 18-0

Telefax +49 (0) 3 51 3 18 18 33

Ad-ak/ak

30. August 2002

5 **Systemonic AG**
01099 Dresden

10 **Verfahren zur Initialisierung von programmierbaren Systemen**

Patentansprüche

- 15 1. Verfahren zur Initialisierung von programmierbaren Systemen bei dem die zur Initialisierung von Registern und internen und/oder externen Baugruppen notwendigen Informationen in einem externen Speicher enthalten sind und ausgelesen werden, insbesondere zur Anwendung in programmierbaren System-on-Chip ASIC Elementen, **dadurch gekennzeichnet**, dass nach dem Einschalten oder einem anderen, einen Neustart auslösenden Ereignis, gesteuert durch ein Programm eines Instruktionsspeichers (9), ein Transfer von Initialisierungsinformationen von einem externen oder internen nichtflüchtigen Speichermedium
- 20 (14) zu einem internen Speicher (10) erfolgt, dass die Initialisierungsinformationen Initialisierungsdaten und/oder mindestens ein Initialisierungsprogramm enthalten, dass die Initialisierung der Register (5) und Baugruppen durch ein oder mehrere im System angeordnete
- 25 Prozessorelemente (8) oder andere intelligente Baublöcke gesteuert erfolgt, die ihrerseits durch das Initialisierungsprogramm gesteuert werden.
- 30
- 35 2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass nach dem Transfer eine Integritätsprüfung der

Initialisierungsinformationen erfolgt und dass, gesteuert durch das Ergebnis der Integritätsprüfung, eine Programmverzweigung durchgeführt wird.

5 3. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass
beim Erkennen eines falschen oder fehlenden internen oder
externen Speichermediums, eine Fehlerroutine abgearbeitet
wird, die die Initialisierung mit Standardwerten
10 durchführt oder den Inhalt des internen oder externen
Speichermediums (14) ganz oder teilweise wiederherstellt.

15 4. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass
die Initialisierungsdaten als Standardwerte aus dem
Speichermedium (14) gelesen werden, durch das
Prozessorelement (8) verändert und die veränderten
Initialisierungsdaten zur Initialisierung verwendet
werden.

20 5. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass
durch das Initialisierungsprogramm von dem
Prozessorelement (8) Initialisierungsdaten berechnet und
zur Initialisierung verwendet werden.

25 6. Verfahren nach Anspruch 5, **dadurch gekennzeichnet**, dass
in Abhängigkeit von Zustandsdaten von
Peripheriekomponenten (7) und/oder internen Komponenten
deren Initialisierungsdaten und die Daten für die
internen Komponenten berechnet werden.

30 7. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass
das Prozessorelement (8) nach erfolgter Initialisierung
in einen Stromsparmodus übergeht.

35 8. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass

15

die Initialisierung weiterer Prozessorelemente (8) gestartet und überwacht wird.

5 9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass eine Anpassung an verschiedene Speichermedien erfolgt.

10 10. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das Initialisierungsprogramm weitere Daten und/oder Programmcode aus einem Speichermedium (14) nachlädt.

10

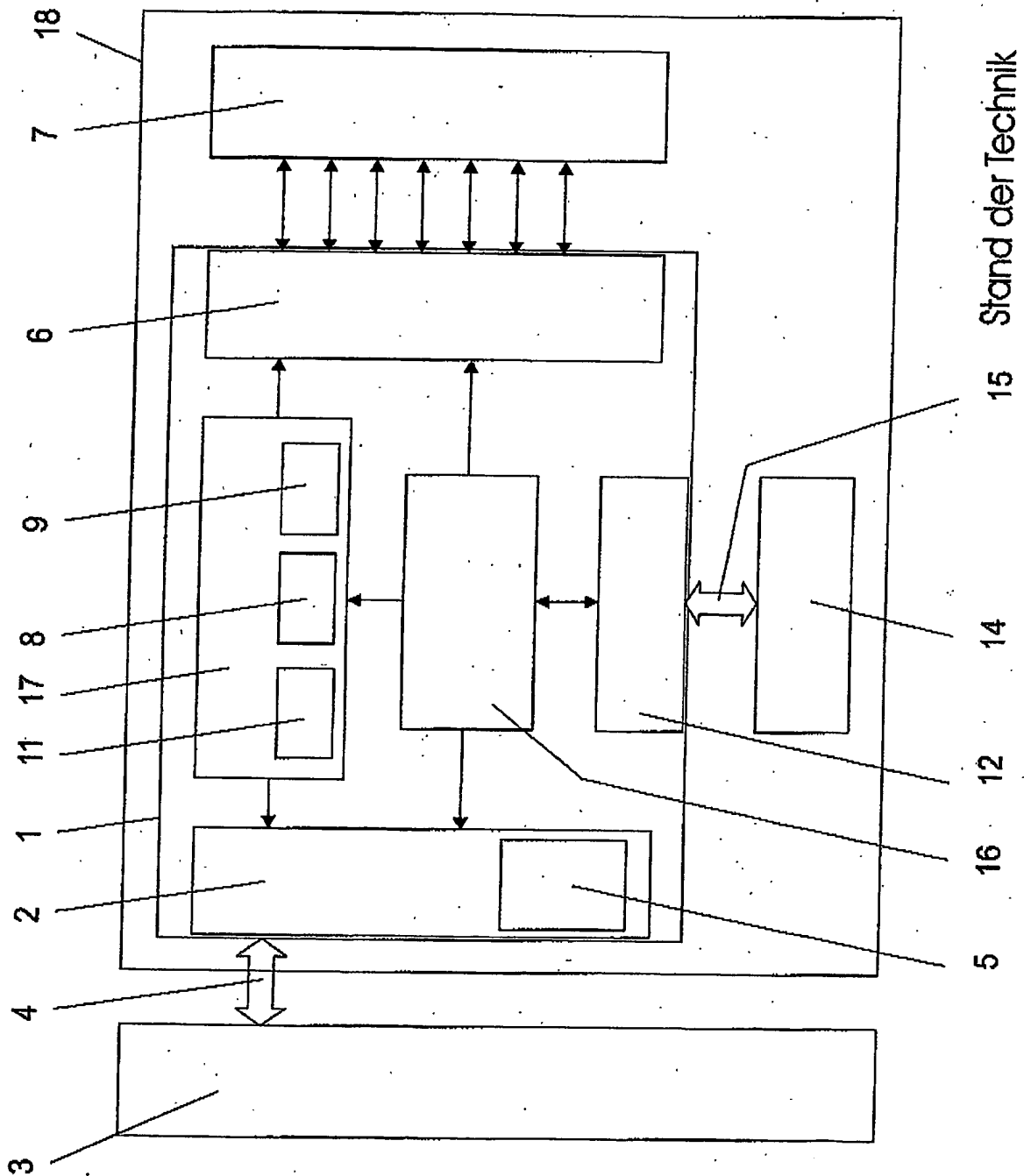


Fig. 1

Stand der Technik

